UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/029,497	12/21/2001	Sivaram Krishnan		4638

7590 07/12/2006

David T. Cunningham, Esq.
Hitachi America, Ltd.
50 Prospect Avenue
Tarrytown, NY 10591

EXAMINER	
GUILL, RUSSELL L	

ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 07/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/029,497	KRISHNAN, SIVARAM	
	Examiner	Art Unit	
	Russ Guill	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 April 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7-10 and 24-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5,7-10,24-31 and 33-35* is/are rejected.

7)☒ Claim(s) *1, 4 and 32-35* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on *21 December 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to an Amendment filed April 7, 2006. Claims 6 and 11 – 23 have been canceled. Claims 32 – 35 have been added. Claims 1 – 5, 7 – 10 and 24 – 35 are pending. Claims 1 – 5, 7 – 10 and 24 – 35 have been examined. Claims 1 – 3, 5, 7 – 10, 24 – 31 and 33 - 35 have been rejected. Claims 1, 4 and 32 – 35 are objected to for minor informalities. Claims 4 and 32 are allowable.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on April 7, 2006 has been entered.

Response to Remarks

3. Regarding claims 1 and 24 rejected under 35 USC § 101:

3.1. Applicant's arguments with respect to claims 1 and 24 have been considered but are moot in view of the new ground(s) of rejection. Upon further search and consideration, an updated rejection has been provided.

4. Regarding new independent claims 32 - 35:

4.1. Upon further search and consideration, rejection is made below for claim 33 - 35.

Claim Objections

5. Claims 1 and 32 - 35 are objected to because of the following informalities: In the claims, lines 8 - 9, the claims recite, "for producing different dynamic execution information." This appears to be an intended use rather than a limitation. Appropriate correction is required.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1 – 3, 10, 24 and 26 - 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Altman (Altman, Erik R.; Kaeli, David; Sheffer, Yaron; "Welcome to the opportunities of Binary Translation", March 2000, IEEE Computer), in view of Mattson (U.S. Patent Number 6,115,809).

7.1. Regarding **claim 1**, the Examiner notes that the although the specification explicitly recites a method that changes the hardware of a computer system dynamically (see page 46, lines 4 – 8 and page 58, lines 6 - 10), the "changing" appears to occur through setting state values of hardware, including setting bits to ones and zeros. The specification does not appear to explicitly recite a method of changing the hardware for

producing different dynamic execution information, and does not appear to explicitly demonstrate that the dynamic optimization methods of the instant invention would produce different dynamic execution information. It appears that producing different dynamic execution information is considered a general side effect of several types of dynamic optimization.

7.2. Regarding **claim 1**, Altman appears to teach:

7.2.1. a method performed by a computer system ***(page 40 – 41, section labeled "Three Types of Translation"; and page 44, left-side column, paragraph that starts with "Translated applications . . .")***.

7.2.2. Obtaining an emulated sequence of instructions derived from an original sequence of instructions ***(page 40 – 41, section labeled "Three Types of Translation")***.

7.2.3. initiating execution of the emulated sequence of instructions (***page 41, left-side column, third paragraph, and section labeled Runtime Optimization; it would have been obvious that the emulated sequence of instructions were executed***).

7.2.4. Producing first dynamic execution information in response to executing the emulated sequence of instructions ***(page 41, section labeled "Profiling", first paragraph)***;

7.3. Altman does not specifically teach:

7.3.1. Changing the hardware dynamically for producing different dynamic execution information in response to said first dynamic execution information.

7.4. Mattson appears to teach:

7.4.1. Changing the hardware dynamically for producing different dynamic execution information in response to said first dynamic execution information (***column 9, lines 49 – 65, and column 10, lines 4 – 28; when the method of column 10, lines 4 – 28 is implemented at runtime as described in column 9, lines 49 – 65, the limitation appears to be satisfied; please note that the phrase "changing the hardware" is broadly interpreted to include setting bits to ones and zeros, which appears to be consistent with the usage in the specification; and column 2, lines 31 - 48***).

7.5. The motivation to use the art of Mattson with the art of Altman would have been the advantages recited in Mattson, including that predictions vary dynamically to changes in the computing environment that may affect branching (***column 3, lines 25 – 31***), and predictions tend to be very accurate when the prediction table is large (***column 3, lines 30 – 33***).

7.6. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Mattson with the art of Altman to produce the claimed invention.

7.7. Regarding **claim 2**, Altman appears to teach:

7.7.1. modifying at least parameters of instructions of the emulated sequence of instructions ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item that starts with "ISA remapping")***.

7.7.2. Regarding ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item that starts with "ISA remapping")***; the registers are parameters of instructions as defined in the specification on page 9, lines 5 – 15.

7.8. Regarding **claim 3**, Altman appears to teach:

7.8.1. modifying at least register fields of instructions of the emulated sequence of instructions ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item that starts with "ISA remapping")***.

7.9. Regarding **claim 10**, Altman appears to teach that changing generates a modified emulated sequence of instructions by modifying at least some instructions of the emulated sequence of instructions in response to at least some of the dynamic execution information ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item and second bullet item)***.

7.9.1. Regarding ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item and second bullet item)***; both the "ISA remapping" and "basic block reordering", modify instructions.

7.10. Regarding **claim 24,** Altman appears to teach:

7.10.1. a storage means storing an emulated sequence of instructions produced from an original sequence of instructions ***(pages 40 – 41, section labeled "Three Types of Translation")***.

7.10.1.1. Regarding ***(pages 40 – 41, section labeled "Three Types of Translation")***; it would have been obvious that storage means are used to store the emulated sequence of instructions (e.g. RAM).

7.10.2. a means for producing dynamic execution information in response to execution of the emulated sequence of instructions ***(page 41, section labeled "Profiling")***.

7.10.3. a processing means for executing the emulated sequence of instructions (***page 41, left-side column, third paragraph, and section labeled Runtime Optimization; it would have been obvious that the emulated sequence of instructions were executed***).

7.11. Altman does not specifically teach:

7.11.1. a means for responding to the dynamic execution information and for changing hardware of the computer system dynamically so that at least some dynamic execution information obtained on subsequent execution of the emulated sequence of instructions would be changed.

7.12. Mattson appears to teach:

7.12.1. a means for responding to the dynamic execution information and for changing hardware of the computer system dynamically so that at least some dynamic execution information obtained on subsequent execution of the emulated sequence of instructions would be changed (***column 9, lines 49 – 65, and column 10, lines 4 – 28; when the method of column 10, lines 4 – 28 is***

implemented at runtime as described in column 9, lines 49 – 65, the limitation is satisfied).

7.13. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Mattson with the art of Altman to produce the claimed invention.

7.14. Regarding **claim 26**, Altman appears to teach modifying at least parameters of instructions of the emulated sequence of instructions ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item that starts with "ISA remapping")***.

7.14.1. Regarding ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item that starts with "ISA remapping")***; the registers are parameters of instructions as defined in the specification on page 9, lines 5 – 15.

7.15. Regarding **claim 27**, Altman appears to teach modifying at least register fields of instructions of the emulated sequence of instructions ***(page 41, section labeled "Dynamic Optimization", second paragraph, first bullet item that starts with "ISA remapping")***.

8. **Claim 5** is rejected under 35 U.S.C. 103(a) as being unpatentable over Altman in view of Mattson as applied to **claims 1 – 3, 10, 24 and 26 - 27** above, further in view of Kistler (Thomas Kistler et al.; "Continuous Program Optimization: Design and Evaluation", June 2001, IEEE Transactions on Computers).

8.1. Altman as modified by Mattson teaches a method of producing dynamic execution information in response to executing emulated instructions, and changing the hardware of a computer system dynamically for producing different dynamic execution information in response to said first dynamic execution information, as recited in **claims 1 – 3, 10, 24 and 26 - 27** above.

8.2. The art of Altman is directed toward translating machine instructions for one computer into machine instructions to run on a second computer, including profiling and dynamic optimization ***(pages 40 - 41)***.

8.3. The art of Kistler is directed to dynamic program optimization ***(page 549, Abstract)***.

8.4. Regarding **claim 5**, Altman does not specifically teach:

8.4.1. said steps of executing, producing and changing are conducted recursively on at least some of successive segments of the emulated sequence of instructions.

8.5. Regarding claim 5, Kistler appears to teach:

8.5.1. steps of executing, producing and changing are conducted recursively on at least some of successive segments of the emulated sequence of instructions (***page 549, Abstract, first three sentences***).

8.6. The motivation to use the art of Kistler with the art of Altman as modified by Mattson would have been the benefits recited in Kistler that the method eliminates some of the most severe performance problems (***page 564, first paragraph***), and can result in real performance improvements (***page 564, third paragraph***).

8.7. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Kistler with the art of Altman as modified by Mattson to produce the claimed invention.

9. **Claim 7** is rejected under 35 U.S.C. 103(a) as being unpatentable over Altman in view of Mattson as applied to **claims 1 – 3, 10, 24 and 26 - 27** above, further in view of Smith (U.S. Patent Number 4,370,711).

9.1. Altman as modified by Mattson teaches a method of producing dynamic execution information in response to executing emulated instructions, and changing the hardware of a computer system dynamically for producing different dynamic execution information in response to said first dynamic execution information, as recited in **claims 1 – 3, 10, 24 and 26 - 27** above.

9.2. The art of Altman is directed toward translating machine instructions for one computer into machine instructions to run on a second computer, including profiling and dynamic optimization ***(pages 40 - 41)***.

9.3. The art of Smith is directed to branch prediction ***(page 1, Abstract)***.

9.4. Regarding **claim 7**, Altman does not specifically teach:

9.4.1. said step of producing, produces branch prediction information;

9.4.2. said step of changing, changes condition codes of the branch prediction information.

9.5. Regarding **claim 7**, Smith appears to teach:

9.5.1. producing branch prediction information (***column 1, lines 44 – 67, and column 2, lines 1 – 2***);

9.5.2. changing condition codes of the branch prediction information (***column 1, lines 44 – 67, and column 2, lines 1 – 2; the branch prediction information is a form of condition code***);

9.6. The motivation to use the art of Smith with the art of Altman as modified by Mattson would have been the benefits recited in Smith including improved branch prediction mechanism with a high prediction accuracy to minimize the time loss caused by incorrect predictions (**column 1, lines 17 -21**), and using less and simpler hardware than another technique might (**column 3, lines 26 – 29**).

9.7. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Smith with the art of Altman as modified by Mattson to produce the claimed invention.

10. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Altman in view of Mattson as applied to **claims 1 – 3, 10, 24 and 26 - 27** above, further in view of Patterson (David A. Patterson et al.; "Computer architecture A Quantitative Approach", second edition, 1996, Morgan Kaufmann Publishers).

10.1. Altman as modified by Mattson teaches a method of producing dynamic execution information in response to executing emulated instructions, and changing the hardware of a computer system dynamically for producing different dynamic execution information in response to said first dynamic execution information, as recited in **claims 1 – 3, 10, 24 and 26 - 27** above.

10.2. The art of Altman is directed toward translating machine instructions for one computer into machine instructions to run on a second computer, including profiling and dynamic optimization ***(pages 40 - 41)***.

10.3. The art of Patterson is directed to common knowledge in the art of computer architecture ***(title)***.

10.4. Regarding **claim 9**, Altman does not specifically teach:

10.4.1. said step of producing, produces a history of branch prediction dynamic execution information;

10.4.2. said step of changing, generates a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group.

10.5. Patterson appears to teach:

10.5.1. producing a history of branch prediction dynamic execution information (***page 262, section Basic Branch Prediction and Branch-Prediction Buffers***);

10.5.2. generating a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group (**pages 266 – 268**).

10.6. The motivation to use the art of Patterson with the art of Altman as modified by Mattson would have been the benefit recited in Patterson of reducing branch penalties (***page 262, section 4.3, title***), which would have been recognized as a valuable benefit by the ordinary artisan.

10.7. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Patterson with the art of Altman as modified by Mattson to produce the claimed invention.

11. Claims 8 and 28 - 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Altman in view of Mattson as applied to **claims 1 – 3, 10, 24 and 26 - 27** above, further in view of Patterson (David A. Patterson et al.; "Computer architecture A Quantitative Approach", second edition, 1996, Morgan Kaufmann Publishers).

11.1. Altman as modified by Mattson teaches a method of producing dynamic execution information in response to executing emulated instructions, and changing the hardware of a computer system dynamically for producing different dynamic execution information in response to said first dynamic execution information, as recited in **claims 1 – 3, 10, 24 and 26 - 27** above.

11.2. The art of Altman is directed toward translating machine instructions for one computer into machine instructions to run on a second computer, including profiling and dynamic optimization ***(pages 40 - 41)***.

11.3. The art of Patterson is directed to common knowledge in the art of computer architecture ***(title)***.

11.4. Regarding **claim 8**, Altman does not specifically teach:

11.4.1. the step of producing produces a history of register allocation information.

11.4.2. the step of changing changes register allocation.

11.5. Regarding **claim 8**, Patterson appears to teach:

11.5.1. producing a history of register allocation information ***(page 233, top half of page)***.

11.5.2. changing register allocation ***(page 233, bottom half of page, and page 232, paragraph that starts with "Both antidependences . . ." recites that the process may be performed dynamically by hardware)***.

11.6. The motivation to use the art of Patterson with the art of Altman would have been the benefit recited in Patterson that resolving dependences allows simultaneous execution of instructions (***page 232, paragraph that starts with "Both antidependences . . ."; and page 229, section Dependences, first paragraph)***, which would have been recognized as a valuable benefit by the ordinary artisan.

11.7. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Patterson with the art of Altman as modified by Mattson to produce the claimed invention.

11.8. Regarding **claim 28**, Altman does not specifically teach:

11.8.1. cycling allocation of registers in a pool of registers as some of successively identified registers in the emulated sequence of instructions.

11.9. Regarding **claim 28**, Patterson appears to teach:

11.9.1. cycling allocation of registers in a pool of registers as some of successively identified registers in the emulated sequence of instructions ***(page 233, bottom half of page)***.

11.10. Regarding **claim 29**, Altman does not specifically teach:

11.10.1. producing a history of temporary register allocation information.

11.10.2. changing register parameters of the emulated sequence of instructions.

11.11. Regarding **claim 29**, Patterson appears to teach:

11.11.1. producing a history of register allocation information ***(page 233, top half of page)***.

11.11.2. changing register parameters of the emulated sequence of instructions ***(page 233, bottom half of page, and page 232, paragraph that starts with "Both antidependences . . ." recites that the process may be performed dynamically by hardware)***.

11.12. Regarding **claim 30**, Altman appears to teach:

11.12.1. an emulation code generator for generating the emulated sequence of instructions that is executable with a first instruction set from the original sequence of instructions that is executable with a different instruction set ***(page 40, title; and abstract directly beneath the title; and left-side column, paragraph 3, definition of "binary translation")***.

11.12.2. modifying the emulated sequence of instructions in response to at least the historical register usage information ***(page 41, section "Dynamic optimization")***.

11.12.3. execution of the emulation sequence of instructions (***page 41, left-side column, third paragraph, and section labeled Runtime Optimization; it would have been obvious that the emulated sequence of instructions were executed***).

11.13. Regarding **claim 30**, Altman does not specifically teach:

11.13.1. generating historical register usage information regarding register status.

11.14. Regarding **claim 30**, Patterson appears to teach:

11.14.1. generating historical register usage information regarding register status during execution ***(page 233, bottom half of page, and page 232, paragraph that starts with "Both antidependences . . ." recites that the process may be performed dynamically by hardware)***.

12. Claims 25 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Altman and Mattson, as applied to **claims 1 – 3, 10, 24 and 26 - 27** above, further in view of Conte (Conte, Thomas M.; Patel, Burzin A.; Cox, J. Stan; "Using Branch Handling Hardware to Support Profile-Driven Optimization", 1994, Proceedings of the 1994 27th annual international symposium on microarchitecture).

12.1. Altman as modified by Mattson teaches a means for producing dynamic execution information in response to execution of an emulated sequence of instructions, and a means for responding to the dynamic execution information and for changing hardware of a computer system dynamically so that at least some dynamic execution information obtained on subsequent execution of the emulated sequence of instructions would be changed, as recited in **claims 1 – 3, 10, 24 and 26 – 27** above.

12.2. The art of Altman is directed toward translating machine instructions for one computer into machine instructions to run on a second computer, including profiling and dynamic optimization ***(pages 40 - 41)***.

12.3. The art of Conte is directed to using branch handling hardware to support profile-driven optimization ***(page 1, Title)***.

12.4. Regarding **claim 25**, Altman does not specifically teach:

12.4.1. maintaining a record of branch addresses in the emulated sequence of instructions historically correlated to whether branches were taken during execution of the emulated sequence of instructions.

12.4.2. means for changing a likelihood condition code of the branch prediction information for at least one of the branches.

12.5. Regarding **claim 25**, Conte appears to teach:

12.5.1. means for maintaining a record of branch addresses in the emulated sequence of instructions historically correlated to whether branches were taken during execution of the emulated sequence of instructions ***(page 2, section 2 "Branch Prediction and Profiling"; and figure 1; and figure 2)***.

12.5.2. means for changing a likelihood condition code of the branch prediction information for at least one of the branches ***(page 2, section 2 "Branch Prediction and Profiling"; and figure 1; and figure 2)***.

12.6. The art of Conte and the art of Altman are analogous art because they both contain the problem of profiling and optimization.

12.7. The motivation to use the art of Conte with the art of Altman would have been the benefit recited in Conte that using the specified branch prediction method produces a dramatic effect in achieving 96% accuracy in branch prediction ***(page 2, section 2.1 Contemporary branch handling mechanisms, second paragraph)***;

12.8. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Conte with the art of Altman as modified by Mattson to produce the claimed invention.

12.9. Regarding **claim 31**, Altman appears to teach:

12.9.1. an emulation code generator for generating the emulated sequence of instructions that is executable with a first instruction set from the original sequence of instructions that is executable with a different second instruction set ***(page 40, title; and abstract directly beneath the title; and left-side column, paragraph 3, definition of "binary translation")***.

12.10. Regarding **claim 31**, Altman does not specifically teach:

12.10.1. generating historical branch prediction dynamic execution information regarding likelihood of branches taken during execution of the emulation sequence of instructions.

12.10.2. generating a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group and is dependent upon a combined effect of the branch predictions of the members of the group.

12.11. Regarding **claim 31**, Conte appears to teach:

12.11.1. generating historical branch prediction dynamic execution information regarding likelihood of branches taken during execution of the emulation sequence of instructions ***(pages 3 – 4, section 3 Using Branch Prediction Hardware for Profiling; and page 2, section 2 Branch Prediction and Profiling)***.

12.11.2. generating a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group and is dependent upon a combined effect of the branch predictions of the members of the group ***(page 2, section 2.1 Contemporary branch handling mechanisms, second paragraph; and figure 2)***.

12.11.2.1. Regarding ***(page 2, section 2.1 Contemporary branch handling mechanisms, second paragraph; and figure 2)***; it would have been obvious to generate a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group and is dependent upon a combined effect of the branch predictions of the members of the group.

13. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Altman (Altman, Erik R.; Kaeli, David; Sheffer, Yaron; "Welcome to the opportunities of Binary Translation", March 2000, IEEE Computer), in view of Mattson (U.S. Patent Number 6,115,809), further in view of Kistler (Thomas Kistler et al.; "Continuous Program Optimization: Design and Evaluation", June 2001, IEEE Transactions on Computers).

13.1. The initial four limitations of claim 33 are taught as recited in claim 1 above. The last limitation is taught as recited in claim 5 above, and repeated below.

13.2. Regarding **claim 33**, Altman does not specifically teach:

13.2.1. said steps of executing, producing and changing are conducted recursively on at least some of successive segments of the emulated sequence of instructions.

13.3. Regarding claim 33, Kistler appears to teach:

13.3.1. steps of executing, producing and changing are conducted recursively on at least some of successive segments of the emulated sequence of instructions (***page 549, Abstract, first three sentences***).

13.4. The motivation to use the art of Kistler with the art of Altman would have been the benefits recited in Kistler that the method eliminates some of the most severe performance problems (***page 564, first paragraph***), and can result in real performance improvements (***page 564, third paragraph***).

13.5. The motivation to use the art of Mattson with the art of Altman would have been the advantages recited in Mattson, including that predictions vary dynamically to changes in the computing environment that may affect branching (***column 3, lines 25 – 31***), and predictions tend to be very accurate when the prediction table is large (***column 3, lines 30 – 33***).

13.6. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Kistler and the art of Mattson with the art of Altman to produce the claimed invention.

14. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Altman (Altman, Erik R.; Kaeli, David; Sheffer, Yaron; "Welcome to the opportunities of Binary Translation", March 2000, IEEE Computer), in view of Mattson (U.S. Patent Number 6,115,809), further in view of Smith (U.S. Patent Number 4,370,711).

14.1. Regarding **claim 34**, Altman appears to teach:

14.1.1. a method performed by a computer system ***(page 40 – 41, section labeled "Three Types of Translation"; and page 44, left-side column, paragraph that starts with "Translated applications . . .")***.

14.1.2. Obtaining an emulated sequence of instructions derived from an original sequence of instructions ***(page 40 – 41, section labeled "Three Types of Translation")***.

14.1.3. initiating execution of the emulated sequence of instructions (***page 41, left-side column, third paragraph, and section labeled Runtime Optimization; it would have been obvious that the emulated sequence of instructions were executed***).

14.1.4. Producing first dynamic execution information in response to executing the emulated sequence of instructions ***(page 41, section labeled "Profiling", first paragraph)***;

14.2. Altman does not specifically teach:

14.2.1. Changing the computer system dynamically for producing different dynamic execution information in response to said first dynamic execution information.

14.2.2. said step of producing, produces branch prediction information;

14.2.3. said step of changing, changes condition codes of the branch prediction information.

14.3. Mattson appears to teach:

14.3.1. Changing the computer system dynamically for producing different dynamic execution information in response to said first dynamic execution information (***column 9, lines 49 – 65, and column 10, lines 4 – 28; when the method of column 10, lines 4 – 28 is implemented at runtime as described in column 9, lines 49 – 65, the limitation is satisfied***).

14.4. Smith appears to teach:

14.4.1. producing branch prediction information (***column 1, lines 44 – 67, and column 2, lines 1 – 2***);

14.4.2. changing condition codes of the branch prediction information (***column 1, lines 44 – 67, and column 2, lines 1 – 2; the branch prediction information is a form of condition code***);

14.5. The motivation to use the art of Mattson with the art of Altman would have been the advantages recited in Mattson, including that predictions vary dynamically to changes in the computing environment that may affect branching (***column 3, lines 25 – 31***),

and predictions tend to be very accurate when the prediction table is large (***column 3, lines 30 – 33***).

14.6. The motivation to use the art of Smith with the art of Altman would have been the benefits recited in Smith including improved branch prediction mechanism with a high prediction accuracy to minimize the time loss caused by incorrect predictions (**column 1, lines 17 -21**), and using less and simpler hardware than another technique might (**column 3, lines 26 – 29**).

14.7. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Mattson and the art of Smith with the art of Altman to produce the claimed invention.

15. Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable over Altman (Altman, Erik R.; Kaeli, David; Sheffer, Yaron; "Welcome to the opportunities of Binary Translation", March 2000, IEEE Computer), in view of Mattson (U.S. Patent Number 6,115,809), further in view of Patterson (David A. Patterson et al.; "Computer architecture A Quantitative Approach", second edition, 1996, Morgan Kaufmann Publishers).

15.1. The initial four limitations of claim 35 are taught as recited in claim 1 above. The last limitation is taught as recited in claim 9 above, and repeated below.

15.2. Regarding **claim 35**, Altman does not specifically teach:

15.2.1. said step of producing, produces a history of branch prediction dynamic execution information;

15.2.2. said step of changing, generates a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group.

15.3. Patterson appears to teach:

15.3.1. producing a history of branch prediction dynamic execution information (***page 262, section Basic Branch Prediction and Branch-Prediction Buffers***);

15.3.2. generating a branch prediction likelihood code for a group of branches that may be different from any branch prediction of the members of the group (**pages 266 – 268**).

15.4. The motivation to use the art of Patterson with the art of Altman would have been the benefit recited in Patterson of reducing branch penalties (***page 262, section 4.3, title***), which would have been recognized as a valuable benefit by the ordinary artisan.

15.5. The motivation to use the art of Mattson with the art of Altman would have been the advantages recited in Mattson, including that predictions vary dynamically to changes in the computing environment that may affect branching (***column 3, lines 25 – 31***), and predictions tend to be very accurate when the prediction table is large (***column 3, lines 30 – 33***).

15.6. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Patterson and the art of Mattson with the art of Altman to produce the claimed invention.

Allowable Subject Matter

16. The indicated allowability of claims 5, 7 and 9 are withdrawn in view of the newly discovered reference(s) as recited in the rejections above.

17. Claim 32 is allowable over the prior art of record.

18. As allowable subject matter has been indicated, applicant's reply must either comply with all formal requirements or specifically traverse each requirement not complied with. See 37 CFR 1.111(b) and MPEP § 707.07(a).

19. Claim 4 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

20. While Altman teaches a method of emulating and optimizing computer instructions including obtaining an emulated sequence of instructions derived from an original sequence of instructions, initiating execution of the emulated sequence of instructions, and producing first dynamic execution information in response to executing the emulated sequence of instructions, and Mattson teaches changing the hardware dynamically for producing different dynamic execution information in response to said first dynamic execution information, neither of these references taken either alone or in combination with the prior art of record teach a method of emulating and optimizing computer instructions specifically including:

20.1. Regarding claims 4 and 32: "said step of changing, includes software producing multiple conditions codes that replace a single condition code of the first dynamic execution information"

In combination with the remaining features and elements of the claimed invention. It is for these reasons that the Applicant's invention defines over the prior art of record.

Conclusion

21. The prior art made of record and not relied upon is considered pertinent to the applicant's disclosure:

21.1. Baxter, U.S. Patent Number 5,794,062, System and method for dynamically reconfigurable computing using a processing unit having changeable internal hardware organization; teaches runtime reconfiguration of computer hardware.

21.2. Guccione, U.S. Patent Number 6,557,156, Method of configuring FPGAS for dynamically reconfigurable computing; teaches runtime reconfiguration of computer hardware.

21.3. Albonesi, U.S. Patent Number 6,205,537, Mechanism for dynamically adapting the complexity of a microprocessor; teaches runtime reconfiguration of computer hardware, especially column 4, lines 5 – 10.

21.4. Brian Fahs et al.; "Performance Characterization of a Hardware Mechanism for Dynamic Optimization", December 2001, Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture; teaches continuous dynamic optimization that changes hardware and subsequent dynamic optimization data.

22. Please note the following reference appears to teach continuous dynamic optimization that changes hardware and subsequent dynamic execution information:

22.1. Thomas Kistler et al.; "Continuous Program Optimization: Design and Evaluation", June 2001, IEEE Transactions on Computers.

23. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday – Friday 10:00 AM – 6:30 PM.

24. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

25. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

RG